rexroth
A Bosch Company

Quick Start Guide

H4U.app Synchronization Control

Function block H4U.app SC - TIA Portal

RE01945-02-Z, 2024-07

Disclaimer

The data specified above only serve to describe the product. As our products are constantly being further developed, no statements concerning a certain condition or suitability for a certain application can be derived from our information. The information given does not release the user from the obligation of own judgment and verification. It must be remembered that our products are subject to a natural process of wear and aging.

Table of contents

1 About this documentation

1.1 About this documentation

This documentation is valid for the following products:

- **H4U**.app Synchronization Control

This Quick Start Guide is valid only in conjunction with the Application Manual “Function block **H4U**.app Synchronization Control" (R.01945-FK).

Read the commissioning instructions and, in particular, the above document completely before working with the function block.

This document describes the function block for a TIA Portal. This function block is therefore referred to as TIA_H4U_app_SC_Synchronization_Control in the following description.

2 Description

2.1 Brief description

The software block **H4U**.app.SC is used for the synchronization control of several hydraulic drives. It offers the following functions:

- Position-synchronized motion of several hydraulic drives
- Synchronized motion in various controller operation modes of the hydraulic drives:
  - Position control: The PI controller synchronizes the positions by correcting the position command values of the hydraulic drives.
  - Velocity control: The PI controller synchronizes the positions by correcting the velocity command values of the hydraulic drives.
  - Alternating motion/force control: The PI controller synchronizes the positions by correcting the force command values of the hydraulic drives that are in closed-loop force control.
- Selection of different reference values for synchronization
- Control of higher-level command value generation
- Model-based following error monitoring

2.2 Interface description

The input and output data of the software block **H4U**.app SC are shown in the following figure and the following table.

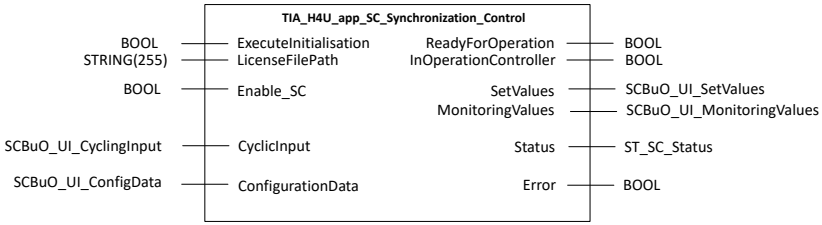

Fig. 1: I/O assignment of the function block

The structures and functionalities used are described in detail in the Application Manual “Function block **H4U**.app Synchronization Control" (R.01945-FK).

VAR_INPUT						
Name	**Type**	**Min**	**Max**	**Def**	**Unit**	**Comment**
ExecuteInitialisation	BOOL	False	True	False	[-]	Enable of license check
LicenseFilePath	STRING(255)	-	-	-	[-]	Path to license file
Enable_SC	BOOL	False	True	False	[-]	Enable of synchronization control
CyclicInput	SCBuO_UI_CyclicInput	-	-	-	[-]	Cyclically required input data (for details, see Application Manual)
ConfigurationData	SCBuO_UI_ConfigData	-	-	-	[-]	Configuration data of synchronization control (for details, see Application Manual)
VAR_OUTPUT						
ReadyForOperation	BOOL	False	True	False	[-]	The initialization of the **H4U**.app has been completed and it is possible to call the functionality of the **H4U**.app.
InOperationController	BOOL	False	True	False	[-]	The synchronization controller is executed.
SetValues	SCBuO_UI_SetValues	-	-	-	[-]	Controller output variables of synchronization control (for details, see Application Manual)
MonitoringValues	SCBuO_UI_MonitoringValues	-	-	-	[-]	Diagnostic values (for details, see Application Manual)
Status	ST_SC_Status	-	-	-	[-]	Status of the app (for details, see Application Manual)
Error	BOOL	False	True	False	[-]	Shows that an error occurred during the execution.

2.3 Functional description

The function block **H4U**.app SC provides functions for the synchronization control of several hydraulic drives. Before the controller functionality can be used, initializing with a license check has to be executed. To this end, first the path to the license file has to be written at input *LicenseFilePath*. The license check is started with a rising edge at input *ExecuteInitialization*. Once the check is completed, the output switches to *ReadyForOperation* = TRUE. Further details can be found in the Application Manual “Function block **H4U**.app Synchronization Control" (R.01945-FK), chapter 5.1.

After successful initialization the controller functionality can be utilized. The function block is parameterized with input *ConfigurationData* and the required actual and command values are applied to input *CyclicInput*. The controller functionality is activated with input *Enable_SC* = TRUE.

As long as the input is set *Enable_SC* = TRUE, license enabling is saved. For a change in licensing, a new license file has to be handed over and initializing has to be restarted with *ExecuteInitialization*. Then, the change is only accepted upon an edge reversal at *Enable_SC*.

The license file may only be read in anew while the system is in a safe state. In the event of an error, output *Error* changes from FALSE to TRUE and the cause of error can be determined from the structure at output *Status*. Invalid component data will not result in an error (for details, see Application Manual).

2.4 Integrating the library

To be able to use the **H4U**.app SC in the TIA Portal, the provided library has to be imported:

1. Activate the license in the Bosch Rexroth licensing portal (for this you require the serial number of the control unit) and download the license file: ➦ https://licensing.boschrexroth.com/

   ℹ For licensing enter the serial number of the control without blank before the year.

2. The license filed is to be stored in the file system in the directory *“UserFiles/“*. The path to the license file then has to be handed over to the **H4U**.app.

   ℹ For some systems, a suitable card reader is required for the memory card used for storing the license file.

   ℹ The file name and the license name can be adapted if you wish to use, for example, a common file path for several projects.

3. A parameter *IndexLastSlaveAxis* is offered under *“PLC tags→ in SelectIndexLastSlaveAxis"*. Here, you have to enter the maximum axis index to be used (number of axes minus 1). It has to be smaller than the number of licensed axes, since otherwise, the function block is not enabled *(ReadyForOperation)*.

4. Integrate the function block in the application project as illustrated in the implementation example. It is recommended that the function block be globally instantiated. Initializing should be carried out in a task with low priority, which can be blocked until initializing is completed. The controller functionality has to be executed in a fast task that meets hard real-time requirements.

5. Configuring the **H4U**.app: To this end, import the pre-configuration from the available data block, adapt it according to the application at hand and transmit it to the input *ConfigurationData*. For details, see Application Manual “Function block **H4U**.app Synchronization Control“ (R.01945-FK).

6. Link values of the input structure *CyclicInput* and control values in the output structure *SetValues* according to the system topology. For details, see Application Manual “Function block **H4U**.app Synchronization Control“ (R.01945-FK).
7. For controller optimization, diagnostics and troubleshooting: Observe notes in the Application Manual “Function block **H4U**.app Synchronization Control“ (R.01945-FK).

2.5 Implementation example

The function block can be integrated as follows:

Global instance block “fbh4uapp_SC“

```
DATA_BLOCK "fbH4Uapp_SC"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
NON_RETAIN
"TIA_H4U_app_SC_Synchronization_Control"

BEGIN

END_DATA_BLOCK
```

Global data block “GLV“

```
DATA_BLOCK "GVL"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
NON_RETAIN
   VAR
      bDoInit : Bool;
      bError : Bool;
      stStatus : "ST_SC_Status";
      sFilePath { S7_SetPoint := 'True'} : String;
      stConfigurationData { S7_SetPoint := 'False'} : "SCBuO_UI_ConfigData";
      stCyclicInput : "SCBuO_UI_CyclicInput";
      udAxLoop { S7_SetPoint := 'True'} : UDInt;
      NumberOfActiveSlaveAxes : UDInt;
      PositionCmdSlave : Array[0.."IndexLastSlaveAxis"] of LReal;
      VelocityCmdSlave : Array[0.."IndexLastSlaveAxis"] of LReal;
      ForceCmdSlave : Array[0.."IndexLastSlaveAxis"] of LReal;
   END_VAR


BEGIN
   sFilePath := 'UserFiles/License.lic';
   udAxLoop := 0;
   NumberOfActiveSlaveAxes := 8;

END_DATA_BLOCK
```

Organization block “Initialization“ for initializing with the license

```
ORGANIZATION_BLOCK "Initialisation"

TITLE = "Main Program Sweep (Cycle)"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Initial_Call : Bool;(*Initial call of this OB*)
```

```
  Remanence : Bool;(*=True, if remanent data are available*)
END_VAR

VAR_TEMP

END_VAR

VAR CONSTANT

END_VAR



BEGIN
  //Detect rising edge
  "fbR_TRIG"(CLK:="GVL".bDoInit);
  //Check if initalisation shall be started
  IF "fbR_TRIG".Q THEN
      "GVL".bDoInit := FALSE;
      "fbH4Uapp_SC".LicenseFilePath := "GVL".sFilePath;
      "fbH4Uapp_SC".ExecuteInitialisation := TRUE;
  END_IF;

  //Call function block only when initialisation is triggered or still active
  IF ("fbR_TRIG".Q OR "fbH4Uapp_SC".Status.StatusLicensing.ActiveInitialisation AND NOT
"fbH4Uapp_SC".ReadyForOperation) THEN
      "fbH4Uapp_SC"();
  END_IF;

  //Reset execute bit after initialisation is done
  IF NOT "fbH4Uapp_SC".Status.StatusLicensing.ActiveInitialisation THEN
      "fbH4Uapp_SC".ExecuteInitialisation := FALSE;
  END_IF;

  //Check for errors
  IF "fbH4Uapp_SC".Error THEN
      "GVL".bError := TRUE;
      "GVL".stStatus := "fbH4Uapp_SC".Status;
  ELSE
      "GVL".bError := FALSE;
  END_IF;
END_ORGANIZATION_BLOCK
```

Organization block “Fast Cycle“ for calling the controller function

```
ORGANIZATION_BLOCK "FastCycle"

TITLE = "Main Program Sweep (Cycle)"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Initial_Call : Bool;(*Initial call of this OB*)
  Remanence : Bool;(*=True, if remanent data are available*)
END_VAR

VAR_TEMP

END_VAR

VAR CONSTANT

END_VAR
```

```
BEGIN
  //Set configuration data and cyclic input
  "fbH4Uapp_SC".ConfigurationData := "GVL".stConfigurationData;
  "fbH4Uapp_SC".CyclicInput := "GVL".stCyclicInput;

  //Call function block when ready for operation
  IF "fbH4Uapp_SC".ReadyForOperation THEN
      "fbH4Uapp_SC"(Enable_SC := TRUE);
  END_IF;

  //get command values
  IF "fbH4Uapp_SC".InOperationController THEN
      FOR "GVL".udAxLoop := 0 TO "GVL".NumberOfActiveSlaveAxes - 1 DO
          "GVL".PositionCmdSlave["GVL".udAxLoop] :=
"fbH4Uapp_SC".SetValues.PositionCmdSlave["GVL".udAxLoop];
          "GVL".VelocityCmdSlave["GVL".udAxLoop] :=
"fbH4Uapp_SC".SetValues.VelocityCmdSlave["GVL".udAxLoop];
          "GVL".ForceCmdSlave["GVL".udAxLoop] :=
"fbH4Uapp_SC".SetValues.ForceCmdSlave["GVL".udAxLoop];
      END_FOR;
  END_IF;

  //Check for errors
  IF "fbH4Uapp_SC".Error THEN
      "fbH4Uapp_SC".Enable_SC := FALSE;
      "GVL".bError := TRUE;
      "GVL".stStatus := "fbH4Uapp_SC".Status;
  ELSE
      "GVL".bError := FALSE;
  END_IF;
END_ORGANIZATION_BLOCK
```

3 Service and support

Contact

Bosch Rexroth AG
Service Industriehydraulik
Bürgermeister-Dr.-Nebel-Straße 8
97816 Lohr am Main
Telefon +49 (0) 9352 405060
Germany
Email: service@boschrexroth.com
Internet: ➦ www.boschrexroth.com/service

For questions about the product

Bosch Rexroth AG
Phone +49 (0) 9352 403020
Email: my.support@boschrexroth.com

Preparation of information

We can help you in a fast and efficient way if you keep the following information available:

- Details about the product concerned, particularly the type code and serial number.
- Your contact details (phone number, e-mail address)

Index

rexroth
A Bosch Company

Bosch Rexroth AG
Zum Eisengießer 1
97816 Lohr a.Main
Germany
Tel. +49 9352 18-0
www.boschrexroth.com

RE01948-02-Z